基于软件定义网络的域内路由保护方案研究 *

张　举，耿海军

(山西大学 软件学院，太原 030013)

摘　要：软件定义网络（SDN）是一种将控制平面和转发平面分离的新型网络体系结构。由于其灵活性和可控性得到了业界的青睐。然而，目前 SDN 采用最优路径转发报文，很难应对网络中频繁出现的节点或者链路故障。因此，为了提高 SDN 网络的可用性，提出了一种基于软件定义网络的域内路由保护方案（intra-domain routing protection scheme based on software defined network，RPBSDN）。该方案可以为网络中的每个源-目的对计算出多个备份下一跳，利用节点加入到最短路径树的偏序关系来保证转发路径没有路由环路。实验结果表明，该方案不仅具有较小的计算复杂度，而且大大提高了网络的可用性。

关键词：软件定义网络；开放最短路径优先；备份下一跳；偏序关系

中图分类号：TP309.7　　doi: 10.3969/j.issn.1001-3695.2017.10.0951

Research on intra-domain routing protection scheme based on software defined network

Zhang Ju, Geng Haijun

(*School of Software Engineering Shanxi University*, *Taiyuan* 03003, *China*)

Abstract: Software defined network (SDN) is a novel network architecture separating control plane and forwarding plane. SDN has been favored by the industry because of its flexibility and controllability. However, SDN usually employs the best paths to forward packets, which is difficult to deal with the node or link failures in the network. In order to improve the availability of SDN network, this paper proposed an intra-domain routing protection scheme based on SDN (RPBSDN) . The scheme could calculate multiple backup next hops for each source-destination, and guaranteed loop-freeness of the induced forwarding path by the underlying partial order of the nodes. The experimental results show that the scheme not only has small computational complexity, but also greatly improve the availability of the network.

Key Words: software defined network; open shortest path first; backup next hop; partial order

0　引言

当前互联网的应用越来越普及，规模也随之增大，各种端系统的应用软件层出不穷，社会对互联网的依赖也越来越深，互联网已经深深的融入大众的日常生活中。但这些端系统的应用都依托于下面的网络层，是在 IP 之上的应用，因此 IP 层的路由设计是非常重要的。特别是某些实时性[1,2]的应用，如 VoIP、股票行情交易软件和在线游戏等，它们对网络的可用性有更严格的要求。

目前，域内的路由都采用动态路由，比较典型的是 OSPF 协议。OSPF 协议是要在源节点和目的节点间找出代价最小的一条路径，以此作为两节点间的路由。但当发生网络故障时，由于改变了域内统一的网络拓扑，需要花费时间完成收敛，并重新计算出路由，这个过程往往需要几秒甚至更多时间，而这对某些实时性要求高的应用来说，是不能满足性能需要的[3]，甚至会造成重大损失。因此，如何在网络出现故障时对路由进行保护，防止网络服务出现中断，成为一个有现实意义的科学问题。

针对网络层的路由保护，学术界提出了一些方案。ECMP（equal cost multiple paths）方案是一种通过管理员调整链路的代价，使得源节点和目的节点间出现多条代价相等的最短路径，以此来达到备份保护路径的目的，但有研究证明是 NP-Hard 问题[4]。LFA（loop free alternates），即无环可选下一跳机制，是基于 IETF 提出的一种快速重路由方案，对应文档为 RFC 5286，由于简单，比较容易布署，该方案已经实现了一定程度的商业化。但其保护程度有限，有研究表明该方案的保护率约为50%。多路由配置方案（multiple routing configurations，MRC）[5]则直接在路由设备中通过增加冗余路由进行路由保护，该方案可处理单节点路由故障，而不需要知道网络状态变化的原因。Not-Via 非逐跳转发方式则用了两种地址，当分组遇到链路故障时，

基金项目：国家“863”计划资助项目（2015AA016105）；国家自然科学基金资助项目（61402253，61702315）

作者简介：张举（1972-），男，山西太原人，讲师，硕士，主要研究方向为路由算法、SDN 网络等（13835193704@139.com）；耿海军（1983-），男，山西太原人，讲师，博士，主要研究方向为网络体系结构、路由算法等.

就将其按照 Not-Via 地址进行封装，绕开节点链路，到达预定节点后，再解封装，并按照 IP 网络的规则进行转发。这种方案计算量较大，也不利于实际部署。

以上研究方案中，或者复杂不易部署，或者需要管理员手动进行配置，也有容易部署但保护效果有限，但所有这些都是分布式的，每个节点要单独计算路由。SDN[6]可以为网络设计者提供了更大的灵活性，更加便于网络管理员配置网络。然而已有的 SDN 依然主要采用最短路径转发报文，当网络中有故障时，仍然需要重新收敛，计算新的路由表，在此期间报文可能会被丢弃。随着 SDN 技术的发展，该技术为解决网络中的故障提供了新的解决思路，相比于传统的网络体系结构，SDN 在实现路由保护方面具有更强的优势[7~9]，有很好的应用前景。欧盟的一项研究深入研究了 SDN 网络中的故障恢复技术。作者在文献[10]中提出一种根据控制器调度的恢复方案，但是该方案具有较高的延迟。作者在文献[11]中提出，在 OpenFlow 中增加 Fast Failover 实现网络故障恢复。相关作者将 SDN 网络中的单故障恢复问题归结化一个整数规划模型[12]，从而利用启发式方法求出近似解。

因此本文所提算法在 SDN 中心节点实现，问题应用环境为域内的路由保护，开创性地将源和目的节点互换，通过一定的算法达到路由保护的目的，而 SDN 中心节点与各数据平面内的通信及流表的生成则不在本文内研究。本文的方案只需要对已有的 SDN 协议做微小的调整就可以实现路由保护，并不会给控制节点带来额外的负担。其思路为在构造以目的为根的树的过程中对节点加入到该树的顺序进行编号，节点之间形成一个偏序关系，利用该偏序关系构造出源节点到目的节点的多条无环路径，从而保证报文转发的正确性。

1　域内 SDN 体系结构设计

在传统的网络体系结构中，网络中的路由设备被设计为控制和数据的统一，路由器既要运算并生成路由，还要实现高速转发。SDN 的思想是将原来网络设备中的控制功能抽离出来，将它们集中到一个控制中心来实现，使数据面和控制面相分离，控制中心则集中于控制运算，而转发设备则专注于转发功能，向上(北向)提供可供开发的编程接口，而向下（南向）则将控制运算结果交给转发设备，形成一个三层体系结构，如图 1 所示。

基础设施层中的设备是通用转发设备，具体使用中的功能受控制层的控制，可以是传统的交换机、路由器、防火墙、流量监控系统等专用设备的功能。转发设备与控制层之间的通信接口使用 Openflow，目前技术已比较成熟。

控制层是一个平台，维护着网络拓扑和一些状态信息等，并将网络控制功能向上提供 API 接口，向下则把控制下达给基础设施层中的通用设备。

应用层进行软件开发，调用控制层的 API，实现网络的各种功能。

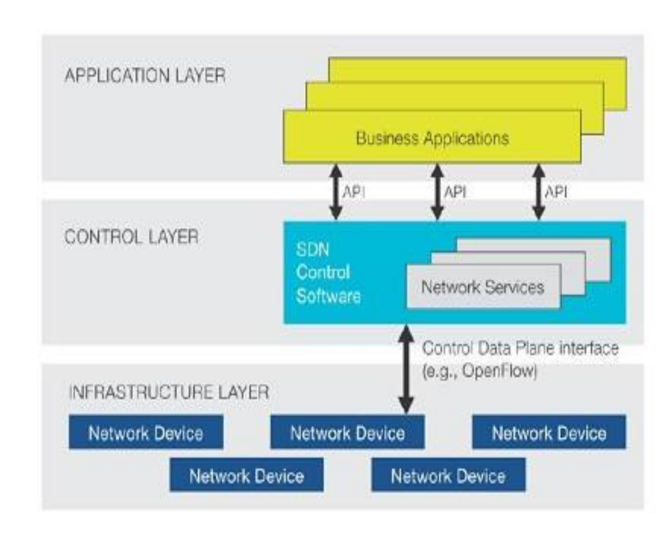


图 1　SDN 体系结构

2　域内路由保护方案及算法

2.1　算法描述

在已经得到域内全网拓扑图的基础上，以目标节点为根构造最短路径树，并对加入到树中的各节点按照加入次序由小到大编号，这样，每个节点可由自身节点开始，寻找编号比它本身编号小的邻居节点构造通往根的路径，而这样的路径可以不是唯一的。由于每个节点都采用这样的算法，就避免了路由环路的产生。

但由于是以目的节点为根构造的最短路径树，所以对某个节点来说，要得到所有 n 个目标节点的路由，就需要这样的 n 棵树。这对于单个节点来说是一个不小的负担，但对 SDN 的体系来说，却发挥了它的集中运算的优点。如此，就可以构造出到所有 n 个目的节点的多条路径，达到路由保护的目的。

2.2　构造以目的节点为根的最短路径树

本文利用最短路径优先算法（shortest path first ，SPF）构造以 v 为根的最短路径树，此处节点 v 为目的节点。算法思路为：从根节点开始寻找距离最短的路径及相邻节点，并将其记入一个集合，称为入选路径集合。再以新入选的节点开始加入新的路径及节点，这个被扩充了的路径和节点集合称为备选路径集。从备选路径集中再寻找一个最短路径，并将其加入入选路径集合，接着再更新备选路径集。重复以上过程，直到所有节点全部加入到入选路径集中为止。

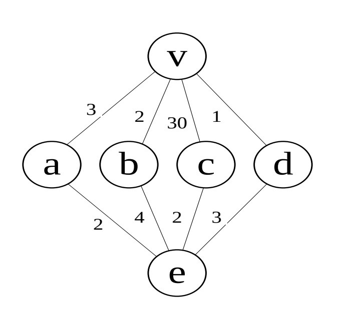


图 2　网络拓扑结构

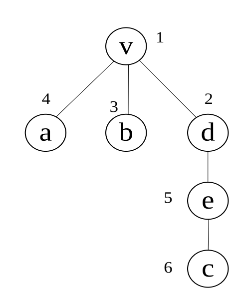


图 3 以节点 v 为根的最短路径树

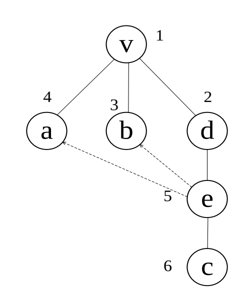


图 4 构造保护路径的例子

2.3 构造保护路径

下面利用一个简单的例子来说明构造保护路径的基本思想。网络拓扑图如图 2 所示，包含 6 个节点和 8 条链路，其中链路中间的数字表示该链路的权值。根据 3.2 节中描述的算法可以构造出以节点 v 为根的最短路径树。如果在构造该树的过程中记录每个节点加入树的顺序，就可以得到图 3。图 3 中的数字表示该节点加入到最短路径树的编号，按加入顺序从小到大编号。

当节点 e 收到目的地址为 v 的数据报时，就从目的地址为 v 的最短路径树中查找下一跳转发地址，查找规则为节点编号比它自身编号小的相邻节点，此处应为 d、b、a 三个节点，如图 4 所示，虚线表示除最优路径外的保护路径。这样，就可以构造出去往目的节点的多条路径。按照编号从小到大的顺序分别为<e,d,v>、<e,b,v>、<e,a,v>。由于编号代表着该节点加入最短路径树的顺序，按照前面最短路径树的构造方法可知，这样的顺序也是一个代价从低到高的顺序，也就是路径优先级从高到低的顺序。

从实际拓扑图来看，节点 e 到节点 v 的路径除了<e,d,v><e,b,v><e,a,v>三条外，还存在其他路径，比如通过节点 c，如果可以这样，那么 c 执行同样的算法还可能会把分组再转发给节点 e，这样就会造成路由环路。下面证明上述方法不会产生路由环路。

2.4 无环路证明

定理 当报文的目的地址为 d 时，节点 c 可以将该报文转发给它的任何一个邻居节点 x，当且仅当节点 c 和节点 x 满足 sequence(d,x)<sequence(d,c)，如果按照上述的条件转发报文，该报文的转发过程将不会出现环路。其中 sequence(d,x)表示在 spt(d)中节点 x 加入到该树的顺序，sequence(d,d)=1，spt(d)表示以节点 d 为根的最短路径树。

证明 假设 $p = (u_n, u_{n-1}, \ldots u_1)$是报文的转发路径，应用上述规则将会得 $sequence(d,u_n)>sequence(d,u_{n-1})>\ldots>sequence(d,u_1)$。因此，可以得出这样的结论：任意两个相邻的节点存在一个严格的偏序关系，因此，该报文的转发过程将不会出现环路。

2.5 算法实现（伪代码）

本节将详细描述算法 RPBSDN 的执行过程。第 1~4 行将所有节点的访问标记属性设置为未访问，代价设置为无穷大，其中 C(c,v) 表示节点 c 到节点 v 的最小代价。第 5~7 行将节点 c 的访问标记属性设置为已访问，代价设置为 0，初始化 seq 的数值。第 8 行将节点 c 加入到队列中。第 10 行选取一个节点出队列。第 11 行计算该节点加入到树中的序列，其中 sequence(c,v) 表示节点 v 加入到 spt(c)中的序列。第 13~14 行更新节点 v 的访问标记和代价。第 15~22 行更新节点 v 的邻居的属性。第 24~30 行计算所有节点到节点 c 的下一跳集合，其中 bn(v,c)表示节点 v 到节点 c 的下一跳集合。

算法 RPBSDN:

输入：网络拓扑结构 G = (V; E)

假设目的地址为 c

```
1 for v∈V do
2       C(c,v) ← ∞;
3       v.visited ← false;
4 endfor
5 c.visited ← true;
6 C(c,c) ← 0;
7 seq← 0
8 Enqueue(Q; < c, 0 >);
9 while Q is not empty do
10      < v, tc >←ExtractMin(Q);
11      sequence(c,v) ←seq;
12       seq++;
13      v.visited ← true;
14      C(c,v) ← tc;
15      for each neighbor u of v do
16          if u.visited = false then
17              newdist ← C (c,v) + L(v; u);
18              if newdist < C(c,u) then
19                  Enqueue(Q; < u, newdist >);
20               endif
21          endif
22       endfor
23 endwhile
24 for   v  ∈  V    do
25     for each neighbor u of v do
26      if   sequence(c,v)>sequence(c,u)
```

```
            bn(v,c)=bn(v,c)∪{u}
        endif
      endfor
ndfor
```

2.6　算法复杂度分析

算法 RPBSDN 主要包含三种优先级队列操作，分别是 Enqueue、ExtractMin、Decrease-Key，本文用 T_e、T_x 和 T_k 来表示三种操作所需要的时间，通过对这三种操作的调用来维护优先级队列 Q。对于操作 Enqueue 和 ExtractMin，算法执行过程中最多调用|V|次，而操作 Decrease-Key 则最多调用|E|次。因此，该算法的复杂度为 $O(|V| * T_e + |V| * T_x + |E| * T_k)$。当使用斐波那契堆来实现优先级队列时，$T_e = O(1)$，$T_x = O(\lg|V|)$，$T_k = O(1)$。因此算法 RPBSDN 的复杂度为 $O(|V| * \lg(|V|) + |E|)$。

3　实验

本章对前面提出的算法 RPBSDN 进行实验模拟，先介绍实验方法，然后通过分析实验结果来说明算法性能。

3.1　实验方法

1）实验拓扑

为了更真实全面的对算法进行评测，这里采用了多种拓扑结构。

（a）美国教育科研网 Abilene[13]，这是美国教育科研网的真实拓扑结构，其中包括 14 条边和 11 个节点。

（b）使用了 Rocketfuel 项目[14]中公开的拓扑，本实验从中选取特征差别比较明显的五个拓扑来作为实验案例，这五个拓扑的节点数量和链路数量分别为 Telstra（108，153）、Ebone（87，162）、Tiscali（161，328）、Exodus（79，147）、Abovenet（141，748）。

（c）利用开源软件 Brite[15]来有目的的生成一些拓扑结构来丰富实验案例，有助于对算法进行较为全面的评测，此处生成的拓扑节点数量为 20~200。

2）实验评价指标

基于本文算法的目的，在此设计两个评价指标，分别是下一跳平均个数和计算效率。在实验中将 RPBSDN 和 SDN-SPF[9]进行比较。SDN-SPF 采用 SDN 方式实现了 OSPF 协议，并且采用最短路径转发报文。

3）实验中，利用一台 PC 机运行 OpenDaylight 控制器来获取网络拓扑结构和计算网络中的路由，另外一台 PC 机运行 Mininet 和 OpenvSwitch，从而可以生成多种多样的拓扑结构。

3.2　下一跳平均个数

下一跳平均个数可以定义为总的下一跳个数/(n*(n-1))，其中 n 为拓扑中的节点个数。从该定义可以知道，下一跳平均个数越多，网络的可用性越高。

图 5 和 6 是利用 Brite 软件生成的拓扑。图 5 中拓扑的节点平均度为 4，对节点数量进行了改变。图 6 中改变了拓扑的节点平均度，保持拓扑中的节点数量不变，均为 200 个。

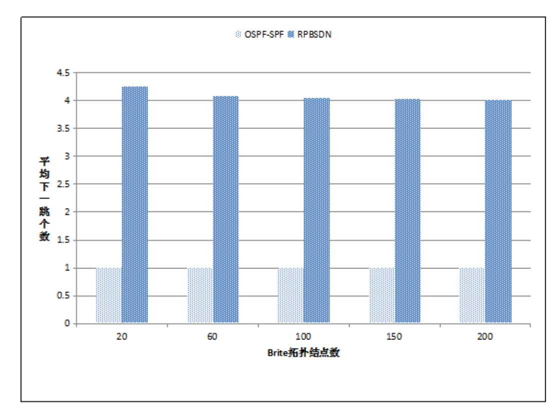


图 5　平均下一跳个数和拓扑大小的关系

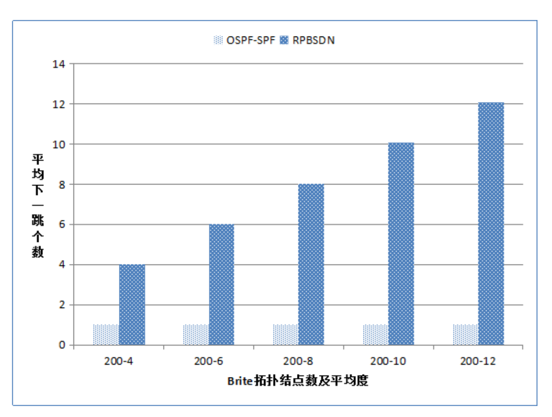


图 6　平均下一跳个数和节点平均度的关系

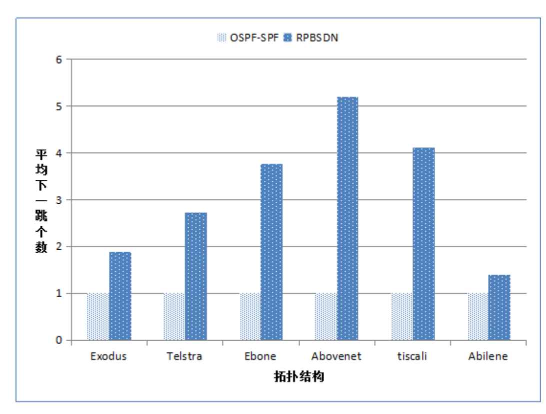


图 7　不同算法在 Abilene 和 Rocketfuel 中的平均下一跳

从图 5、6 可以看出，节点中路由的平均下一跳个数与其平均度相近，这主要是因为用 Brite 软件生成的拓扑较规则，连通度较好。相比较 OSPF-SPF 算法来说，RPBSDN 路由有更多的选择，达到了路由保护的目的。图 7 中的拓扑选取了美国教育科研网 Abilene 和 Rocketfuel 项目中公开的几个拓扑。从图中可以看出，Abilene 的平均下一跳为 1.4，数值最小；Abovenet 的平均下一跳最大，为 5.21，主要原因在于其网络拓扑的连通度不同。连通度越大，平均下一跳个数也越多，符合 RPBSDN 预期。

3.3　计算效率

为了消除不确定因素的影响，计算效率用构造最短路径树的次数来模拟。从图 8 来看，在所有拓扑结构中 OSPF-SPF 和 RPBSDN 的计算效率都是一样的。这是因为在 SDN 中，控制

器需要为所有节点计算路由表，如果将 RPBSDN 部署在 SDN 中，控制器需要执行|V|次 RPBSDN 算法，所以复杂度为|V|O(|V| * lg(|V|) + |E|)，同样 OSPF-SPF 也需要为网络中所有节点计算路由表，控制器需要执行|V|次 SPF 算法，其复杂度同样为|V|O(|V| * lg(|V|) + |E|)。所以，如果在 SDN 中实现 OSPF-SPF 和 RPBSDN，其复杂度是一样的。

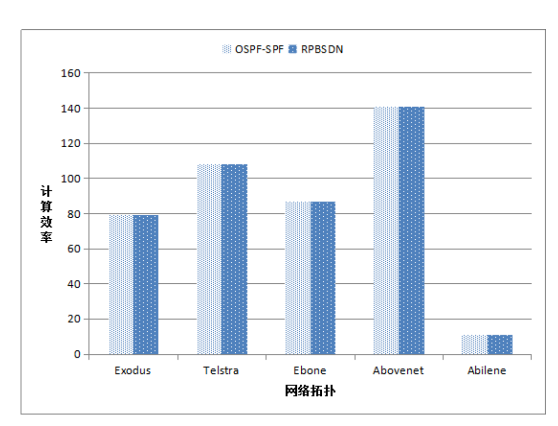


图 8 不同算法在 Abilene 和 Rocketfuel 中的计算效率

4 结束语

因特网中，域内路由协议多采用 OSPF，由于该协议使用了 Dijkstra 提出的最短路径算法 SPF，所以针对某一目的地，下一跳只取最短路径。这样，当链路出现故障时，直到新的路由生成，网络通信处于不正常状态。RPBSDN 可以针对某一目的地生成多条路由，增加了网络的可靠性，形成路由保护。

另外，鉴于 SDN 应用的快速发展，本文将算法放在 SDN 中心节点来实现，依靠 SDN 中心的强大计算能力，能更加发挥 RPBSDN 的优点。但限于环境等因素，本文在实验中对 SDN 环境的模拟还不够真实，这也是下一步要做的工作。

参考文献：

[1] 李清. 基于弱转发的互联网路由可用性和扩展性研究 [D]. 北京：清华大学, 2013.

[2] Zheng J, Xu H, Zhu X, et al. We've got you covered: failure recovery with backup tunnels in traffic engineering [C]// Proc of IEEE International Conference on Network Protocols. 2016: 1-10.

[3] 耿海军. 基于路由度量的域内多路径路由研究 [D]. 北京：清华大学, 2015.

[4] Sridharan A, Guerin R, Diot C. Achieving near-optimal traffic engineering solutions for current ospf//is-is networks [J]. IEEE//ACM Trans on Networking, 2005, 13 (2): 234-247.

[5] Kvalbein A, Hansen A F, Cicic T, et al. Fast IP network recovery using multiple routing configurations [C]// Proc of IEEE International Conference on Computer Communications. 2007: 1-11.

[6] 张朝昆，崔勇，唐翯祎，等. 软件定义网络 (SDN) 研究进展 [J]. 软件学报, 2015, 26 (1): 62-81.

[7] Sharma S, Staessens D, Colle D, et al. Fast failure recovery for in-band OpenFlow networks [C]// Proc of the 9th International Conference on Design of Reliable Communication Networks. 2013: 52-59.

[8] Hartert R, Vissicchio S, Schaus P, et al. A declarative and expressive approach to control forwarding paths in carrier-grade Networks [J]. ACM Sigcomm Computer Communication Review, 2015, 45 (5): 15-28.

[9] Diego Kreutz, Ramos F M V, Veríssimo P E, et al. Software-defined networking: a comprehensive survey [J]. Proceedings of the IEEE, 2015, 103 (1): 14-76.

[10] Sharma S, Staessens D, Colle D, et al. Enabling fast failure recovery in OpenFlow networks [C]// Proc of the 8th International Workshop on Design of Reliable Communication Networks. 2011: 164-171.

[11] Sharma S, Staessens D, Colle D, et al. OpenFlow: Meeting carrier-grade recovery requirements [J]. Computer Communications, 2013, 36 (6): 656-665.

[12] Hao F, Kodialam M, Lakshman T V. Optimizing restoration with segment routing [C]// Proc of the 35th Annual IEEE International Conference on Computer Communications. 2016: 1-9.

[13] Advanced networking for research and education. [EB/OL]. https://www. internet2. edu/products-services/advanced-networking.

[14] Spring N, Mahajan R, Wetherall D, et al. Measuring ISP topologies with rocketfuel [J]. IEEE//ACM Trans on Networking, 2004, 12 (1): 2-16.

[15] http://www. cs. bu. edu/brite [EB/OL].